



Samueli
School of Engineering

Agent Society Challenge

Final Presentation: User Simulation

Team 12: Evan , Rita , Abby   , Kelly 

Dec 3, 2025

Background

Motivation

Recent advances in large-language model (LLM) based agents have opened up new possibilities for simulating real-world user behaviors.

Goal

Develop an agent to simulate user behavior, including generating reviews and ratings.

Evaluation Metrics

Preference Estimation:

- Reflects MAE

Review Generation:

- Reflects sentiment, emotion, and topic error

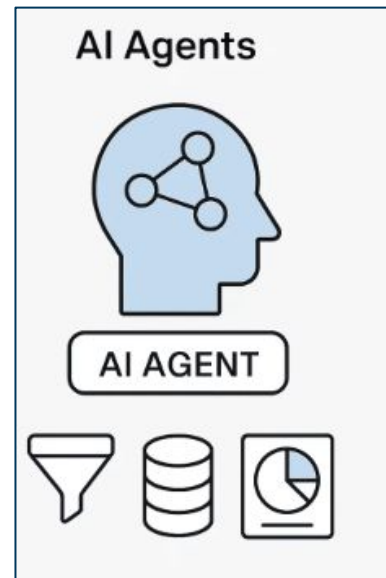
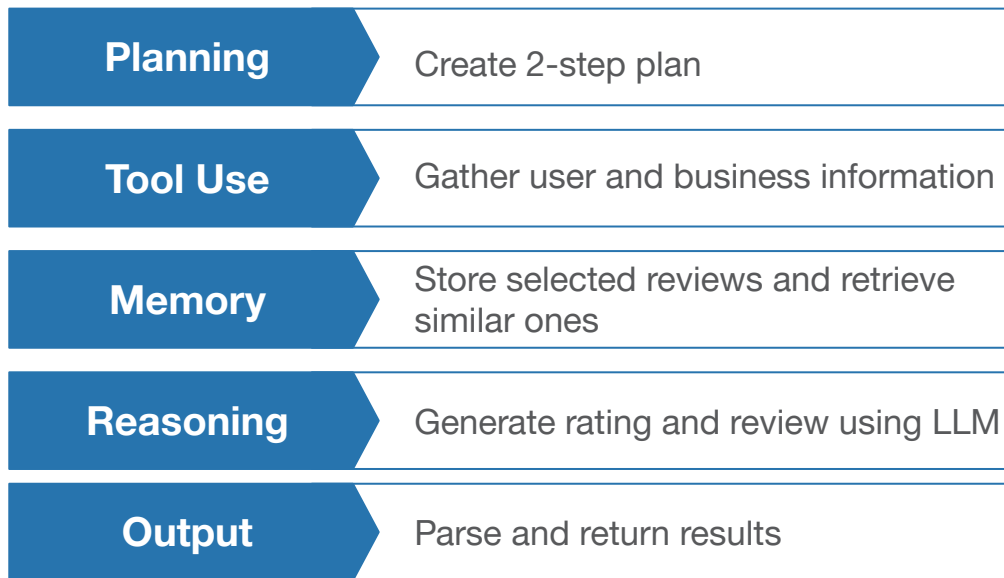
Overall Quality:

- Average Score of previous two

Configuration

Dataset: Yelp (item, review, user)

Baseline Model:



Implementation Decisions

- Chain-of-Thought prompting
- Iterative Retrieval via query expansion
- Selective review filtering before memory storage
- User pattern analysis for style enforcement

Chain-of-Thought prompting

1. Structured 3-Step Reasoning Process:

Guides the LLM through explicit steps: (1) Analyze user's rating behavior, (2) Assess business reputation, (3) Match user tendency with business quality to determine rating

2. Decision Matrix for Rating Prediction:

Provides explicit logic mapping user type \times business quality \rightarrow expected rating (e.g., "Generous reviewer + High-rated business \rightarrow 5.0 stars")

3. Sentiment-Rating Alignment:

Enforces consistency between star rating and review text (high ratings \rightarrow positive language, low ratings \rightarrow critical language)

4. Results:

Improved overall quality from baseline to 0.86-0.88 by helping the LLM reason about user preferences rather than pure, unguided, pattern-matching

Iterative Retrieval via query expansion

1. Iterative Retrieval Process

- a. Begin with initial query
- b. Using an LLM, generate new expanded queries using initial query
- c. Retrieve similar reviews from memory using expanded queries

2. Baseline

Only uses initial query to retrieve similar reviews from memory.

3. Improvement

Using iterative retrieval, via query expansion, we enhance the memory retrieval process by expanding the initial query into multiple semantically related queries. Iterative retrieval captures a broader context related to the user's review style and content.

Selective review filtering before memory storage

1. Solves Context Length Problem

Baseline: Stores all business reviews (some have quantity with 100+)

Smart selection: Only selects top-15 most relevant reviews

Impact: Prevents exceeding LLM context limits and reduces noise

2. User-Centric Selection

Baseline: Stores all reviews without considering user characteristics

Smart selection: Uses average embedding of user's reviews as query

Impact: Selects reviews more similar to the user's writing style

3. Weighted Average Embedding

Baseline: Not applicable

Smart selection: Weighted by rating

Impact: Emphasizes the user's highly-rated reviews for a more accurate average embedding

User pattern analysis for style enforcement

1. User Persona Extraction

Analyzes raw user history to profile a user's specific voice by calculating **Verbosity** (sentence length consistency) and **Vocabulary** (recurring signature phrases) to prevent generic writing styles.

2. The Average Rating Problem

Initial experiments enforcing statistical average ratings caused performance regression. Hard statistical averages masked nuanced, **bimodal behaviors** (e.g. users who only rate 1 or 5 stars) that the baseline detected better intuitively.

3. Decoupled Style Enforcement Strategy

Adopted a hybrid architecture to solve the regression: The baseline handles **Star Rating** prediction while the user pattern analysis module strictly enforces **Writing Style**.

4. Results

Successfully improved **review quality** by enforcing user-specific vocabulary and length constraints without degrading **rating accuracy**, validating the decoupled approach.

Experimentation

- Run ablation studies (with / without each module)
- Compare against baseline
- Optimize prompts and combine strategies

Experimentation

Results

Trial	#	Preference Estimation	#	Review Generation	#	Overall Quality	#	Count
AEKR		0.854		0.8512583334		0.8526291667		50
EKR		0.85		0.8689542508		0.8594771254		50
AEK		0.854		0.870355825		0.8621779125		50
AER		0.842		0.8619831793		0.8519915896		50
AKR		0.85		0.8524477937		0.8512238968		50
AE		0.85		0.8657335099		0.857866755		50
AK		0.83		0.8523652587		0.8411826294		50
AR		0.842		0.8490504888		0.8455252444		50
EK		0.842		0.8672476079		0.854623804		50
ER		0.858		0.8715911428		0.8647955714		50
KR		0.842		0.866940981		0.8544704905		50
A		0.838		0.8367388306		0.8373694153		50
E		0.85		0.8644330946		0.8572165473		50
R		0.834		0.8479675446		0.8409837723		50
K		0.81		0.8504545073		0.8302272537		50
Baseline		0.838		0.8456909738		0.8418454869		50
Baseline		0.822		0.830344303		0.8261721515		50
Baseline		0.83		0.8280888252		0.8290444126		50

A: User pattern analysis for style enforcement
K: Iterative Retrieval via query expansion
E: Chain-of-Thought prompting
R: Smart Context Selection

1st

2nd

3rd

Performance

Evaluation Metrics	Baseline	Best Agent [CoT + SCS]	Improvement
Preference Estimation	0.822	0.858	+4.37%
Review Generation	0.830	0.872	+5.06%
Overall Quality	0.826	0.865	+4.72%

Conclusion

Our experiment indicates that simulation-based data generation can effectively supplement real user data for tasks under appropriate conditions. Compared to baseline approaches, our approach achieves higher accuracy. Improving memory retrieval, LLM prompting, and past user behavior analysis are some of the proven methods to improve simulation performance. Future work may focus on enhancing the memory module to better interact with the LLM, extending the capability to different platforms, as well as exploring more strategies to allow the model to reflect and improve on its own outputs.

Q&A
